Статический анализ в open-source-проектах на примере разработки ClickHouse

Георгий Грибков





Докладчик

Георгий Грибков

Программист C++, один из разработчиков статического анализатора кода PVS-Studio

Автор статей об использовании статического анализа
в проектах с открытым
исходным кодом

gribkov@viva64.com



Содержание

1. Статический анализ: что это такое
2. Интеграция в разработку открытых проектов
3. Использование PVS-Studio в ClickHouse
4. Примеры найденных в ClickHouse ошибок
5. Заключение



Статический анализ: что это такое

Статический анализ – это…

- …это проверка кодовой базы с помощью специального ПО
- …это code review в автоматическом режиме



Что может обнаружить стантанализ

- Отклонения от code style
- Code smells
- Нарушения стандарта кодирования (MISRA, AUTOSAR C++ и т.д.)
- Опечатки, баги, copy-paste
- Потенциальные уязвимости (CWE, CERT)



Почему это необходимо?

Относительная цена исправления ошибки
в зависимости от времени её обнаружения

Этап	
30x	
25x	
20x	
15x	
10x	
5x	
0x	

Проектирование
Разработка
Компонентное/ интеграционное тестирование
Системное/ приёмочное тестирование
Пост-релиз



Почему это необходимо?

30x
25x
20x
15x
10x
5x
0x

Относительная цена исправления ошибки
в зависимости от времени её обнаружения

Статанализ

Проектирование
Разработка
Компонентное/
интеграционное
тестирование
Системное/
приёмочное
тестирование
Пост-релиз



Современные статические анализаторы

- PVS-Studio
- ReSharper
- Coverity
- SonarQube
- Klocwork
- Clang Static Analyzer
- IntelliJ IDEA
- …

- Большой список статических анализаторов:







Интеграция в разработку открытых проектов

Про интеграцию вообще

1. Классический сценарий интеграции
2. Проблема интеграции в открытые проекты…
3. …и её решение



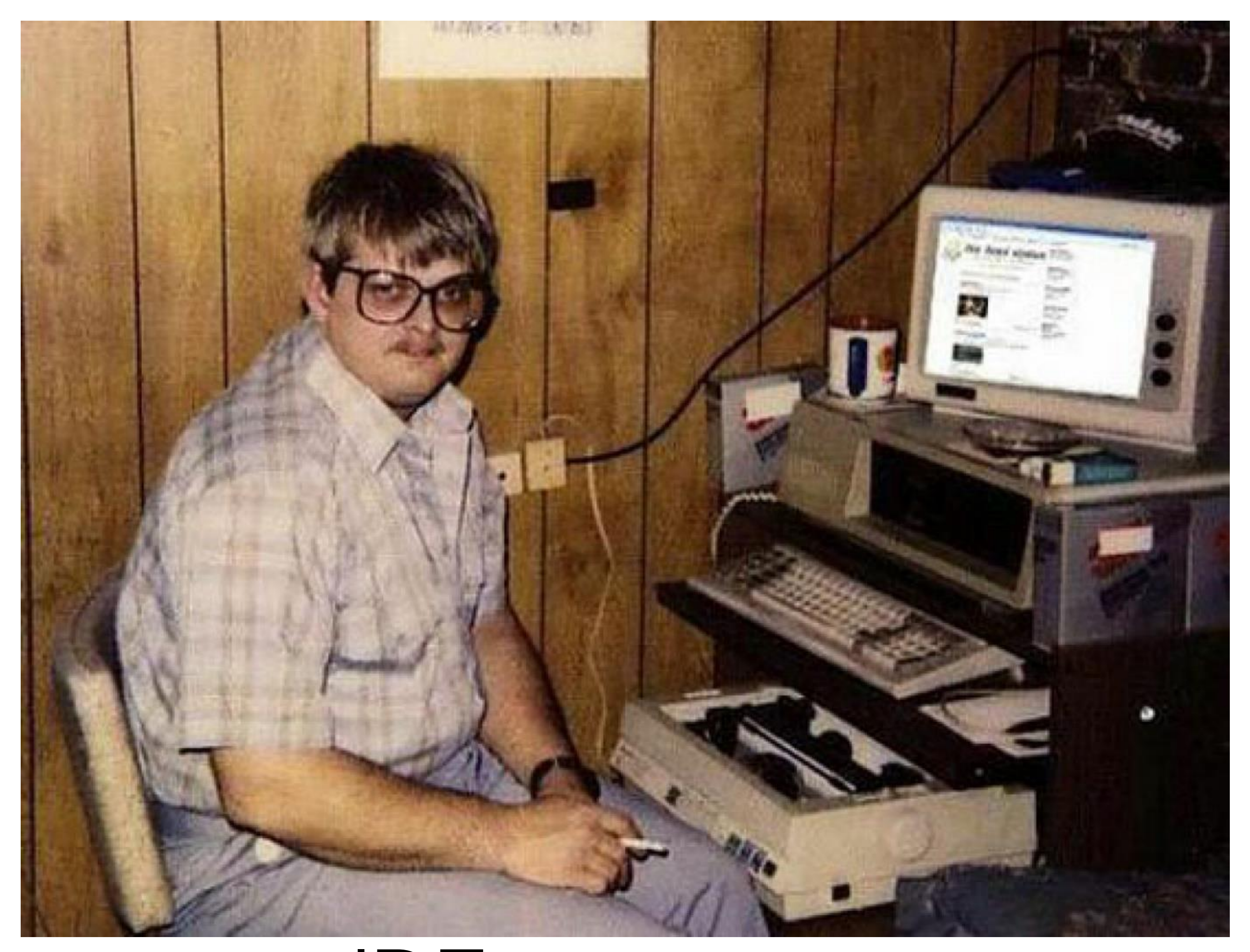
Классический сценарий интеграции

Основные принципы:

- Анализ локально
- Анализ на сервере

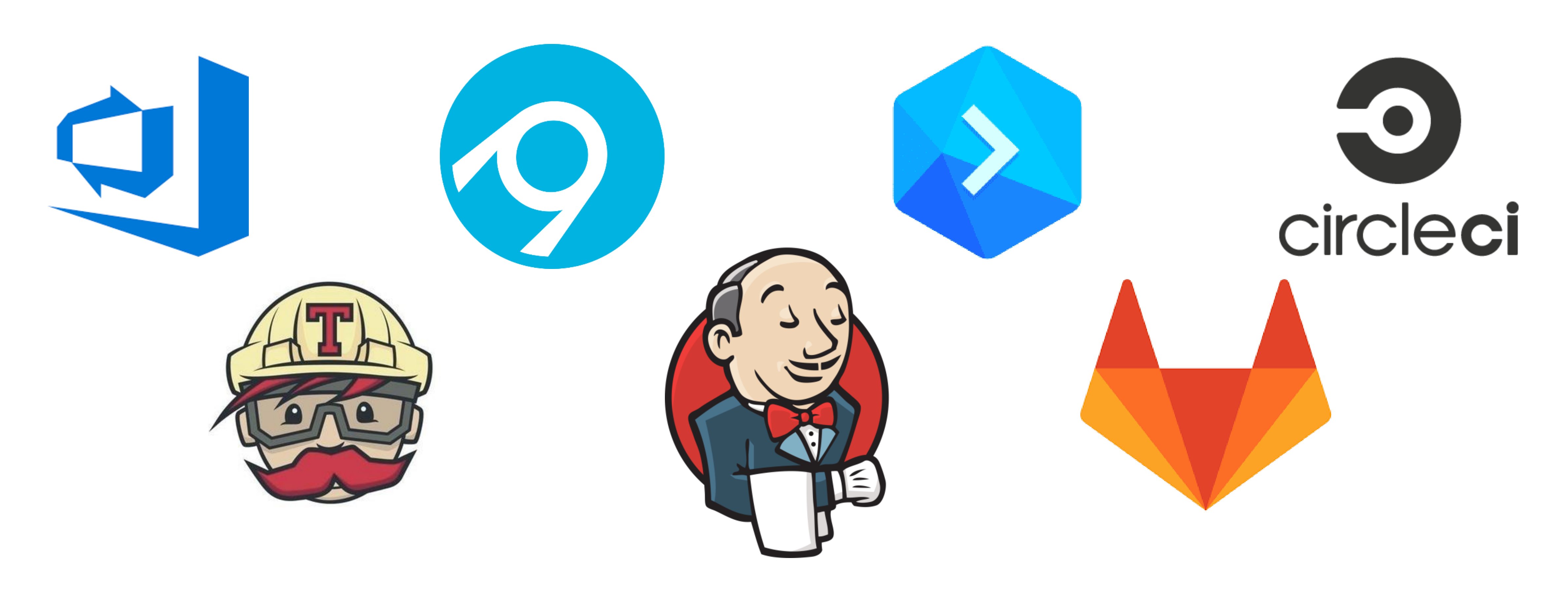


Анализ на компьютерах разработчиков



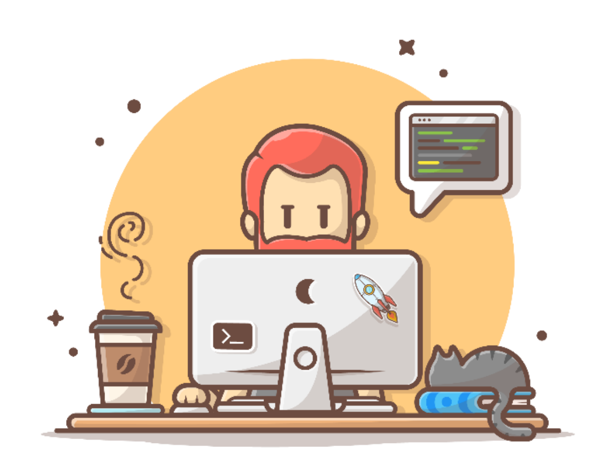
(плагины для IDE, системы мониторинга компиляции и т.д.)

Автоматический анализ на сервере



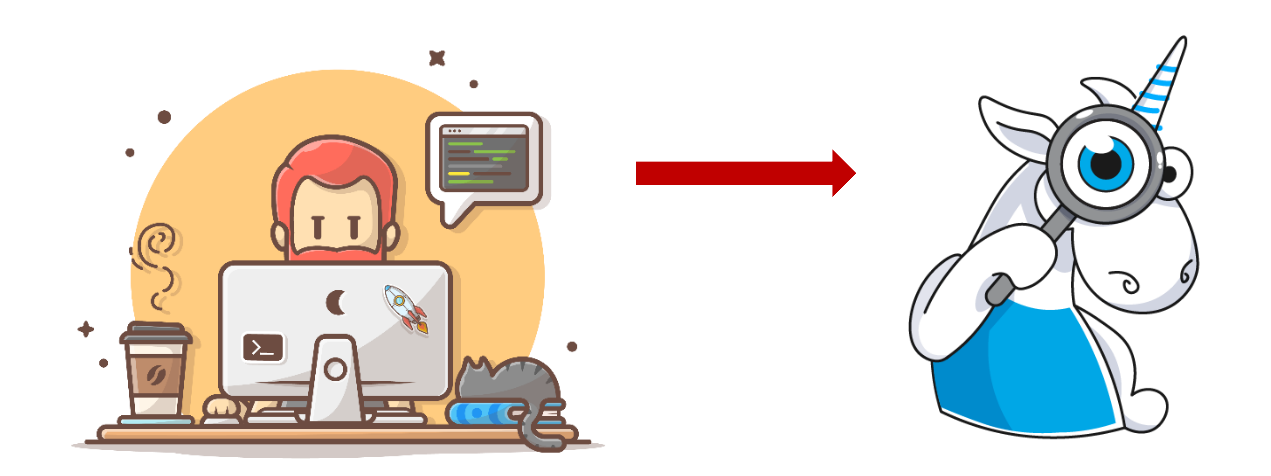
(command-line-утилиты, плагины для CI-систем, системы мониторинга)

Классический сценарий интеграции



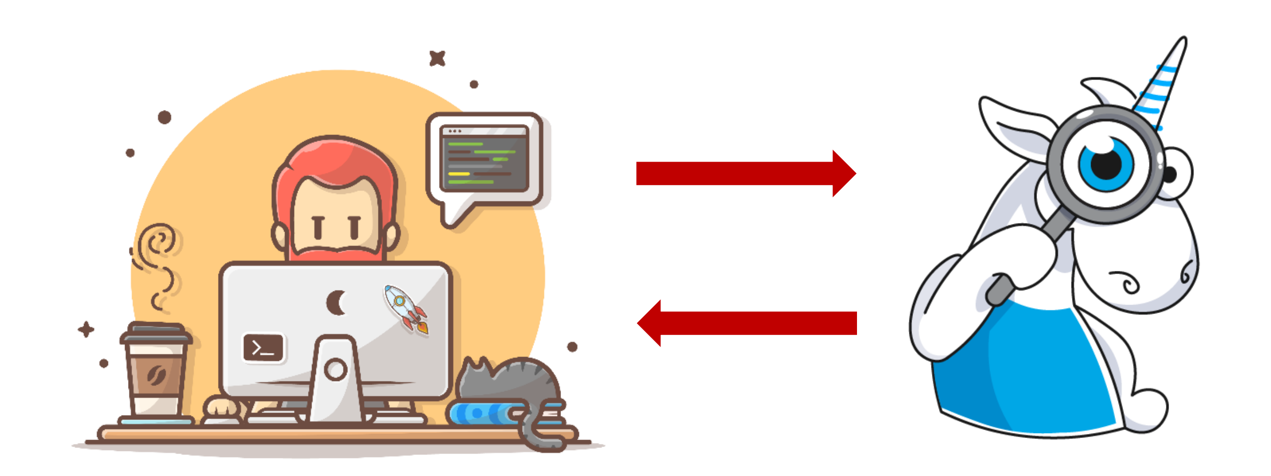


Классический сценарий интеграции



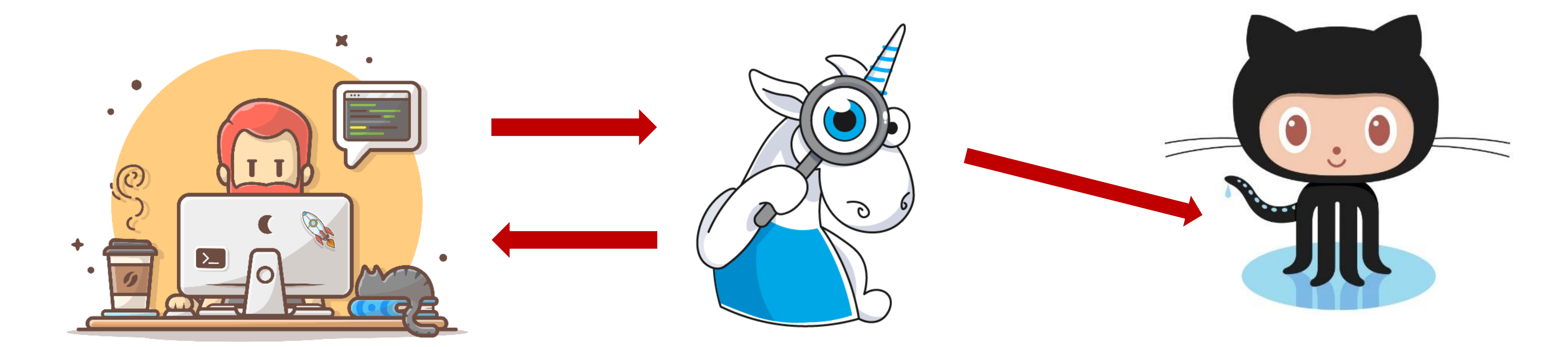


Классический сценарий интеграции





Классический сценарий интеграции





Классический сценарий интеграции



Классический сценарий интеграции



Классический сценарий интеграции



Классический сценарий интеграции



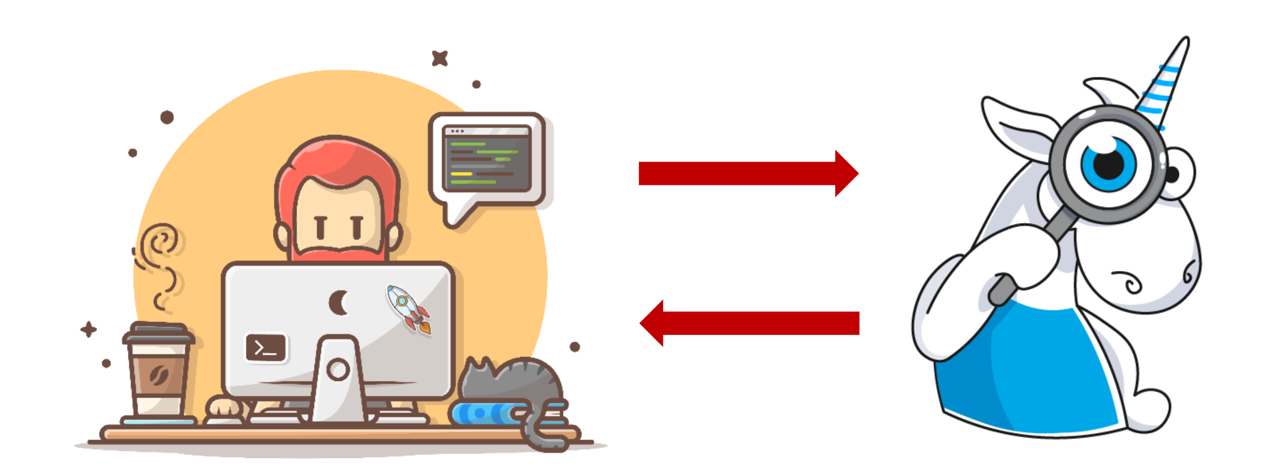
Классический сценарий интеграции



Классический сценарий интеграции



Классический сценарий интеграции





Классический сценарий интеграции



Применение анализа в open-source

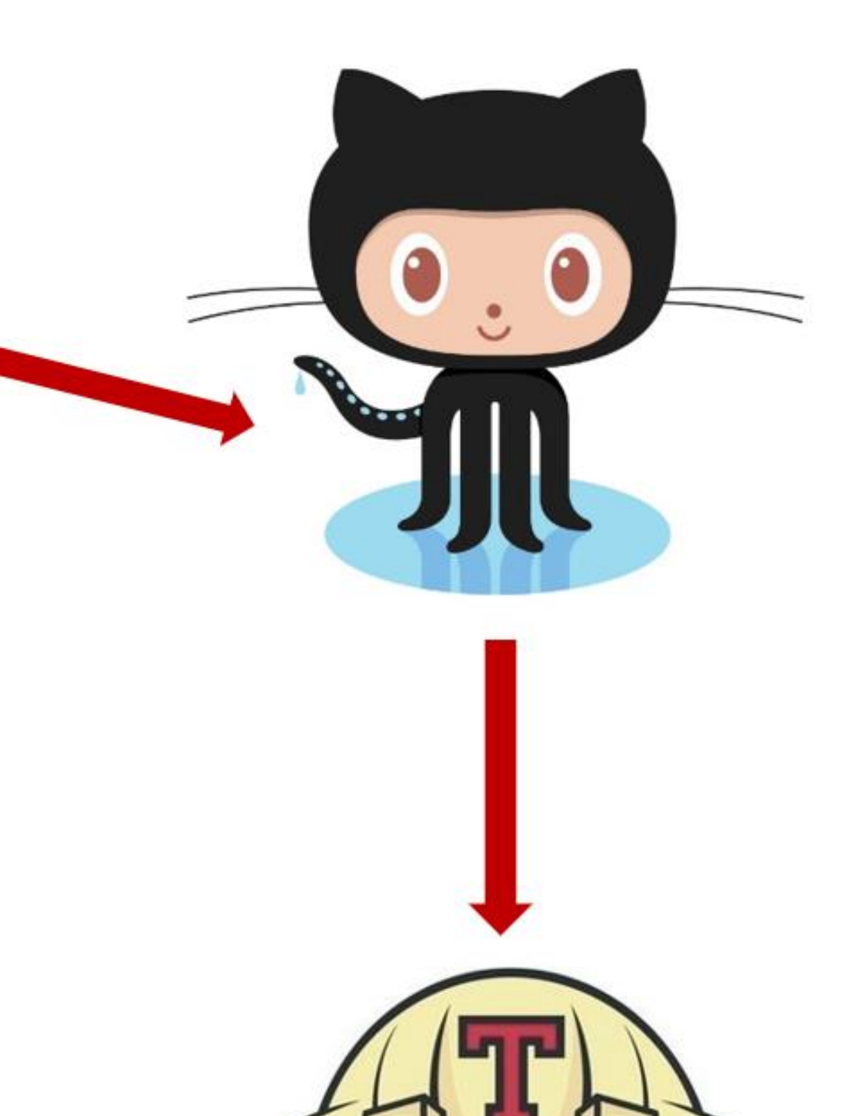
В ЧЕМ ОТЛИЧИЕ?



Интеграция в открытые проекты

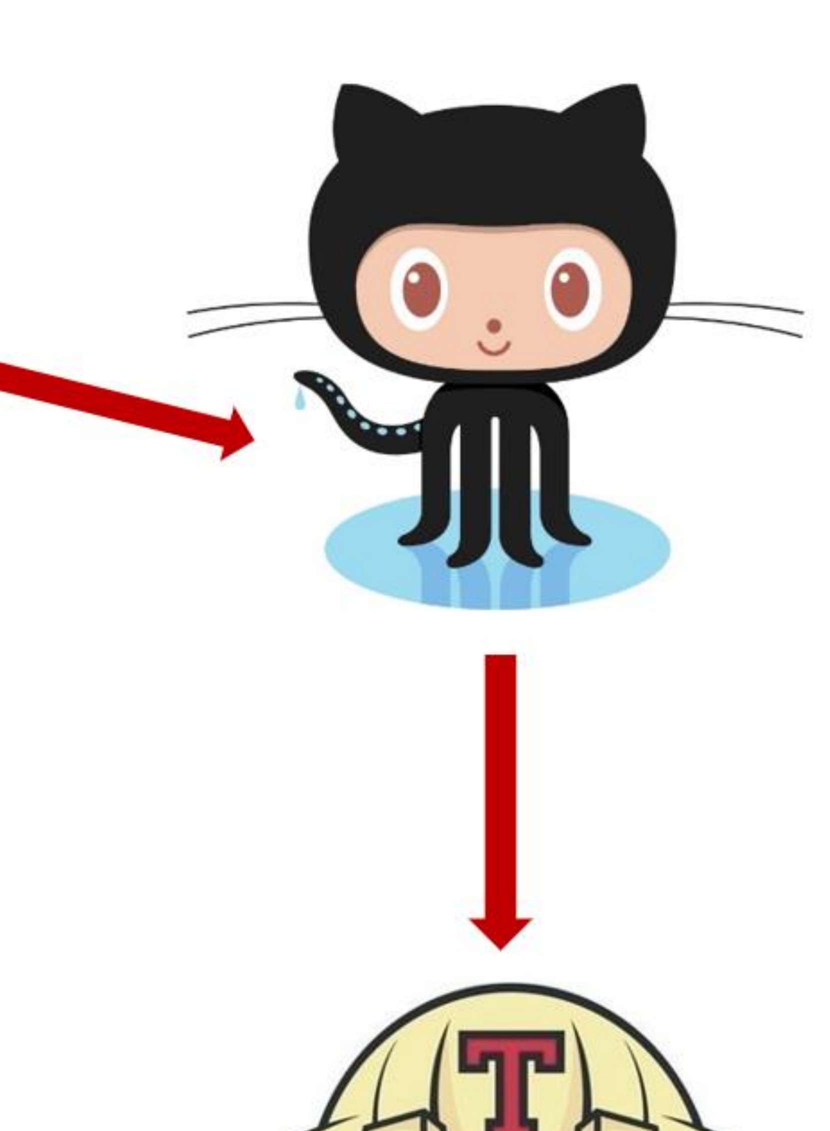


Интеграция в открытые проекты





Интеграция в открытые проекты







Интеграция в открытые проекты





Интеграция в открытые проекты



Как анализировать вклад сообщества?



Анализ коммитов

- Вариант 1: инкрементальный анализ.
- Вариант 2: прямое сравнение логов.



Использование в ClickHouse

С чего все началось

Бесплатная лицензия PVS-Studio для открытых проектов: www.viva64.com/pvs-free-opensource





Специфичные моменты при анализе ClickHouse

- Собирается на Ubuntu
- Имеет сообщество
- Хочется наглядности



Что решено было делать

- Для анализа на Linux – трассировка компиляции
- Для коммитов – прямое сравнение логов (вариант 2)
- Для наглядности результатов – загрузка логов на GitHub



Что решено было делать

CI



Трассировка компиляции

- Очень просто
- Использует strace_out
- Не зависит от сборочной системы

- `pvs-studio-analyzer trace -- ...`
- `pvs-studio-analyzer analyze [OPTIONS]`



Конвертация и сохранение логов

- plog-converter
- Публичный бакет S3
- GitHub API



Add an SQL function formatRead ×

github.com/ClickHouse/ClickHouse/pull/15497/commits

Open **Add an SQL function formatReadableTimeDelta to format time delta** #15497

filipecaixeta wants to merge 5 commits into ClickHouse:master from filipecaixeta:master

add maximum_unit arg to formatReadableTimeDelta function
filipecaixeta committed 6 days ago ✗ — 146f973

fix style and pvs check
filipecaixeta committed 6 days ago ✗ — bab7c76

fix style and pvs check
filipecaixeta committed 6 days ago ✗ — 0436e0f

Commits on Oct 8, 2020

fix pvs check
filipecaixeta committed 14 hours ago ✗ — 3861a03

Some checks were not successful

2 failing, 5 pending, and 8 successful checks

Status	Check	Result	
✗	PVS check	Found 2 new errors, total 30 errors	Details
✗	Style check	Found 2 errors	Details
●	ClickHouse build check	*Pending — 7/16 builds fini...*	Details
●	ClickHouse special build check	*Pending — 4/5 bui...*	Details
●	Functional stateful tests (debug)	*Pending — Started*	Details
●	Functional stateless tests (debug)	*Pending — Star*	Details

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About



Add an SQL function formatRead ×

github.com/ClickHouse/ClickHouse/pull/15497/commits

Open Add an SQL function formatReadableTimeDelta to format time delta #15497

filipecaixeta wants to merge 5 commits into `ClickHouse:master` from `filipecaixeta:master`

add maximum_unit arg to formatReadableTimeDelta function
filipecaixeta committed 6 days ago ✕ — 146f973

fix style and pvs check
filipecaixeta committed 6 days ago ✕ — bab7c76

fix style and pvs check
filipecaixeta committed 6 days ago ✕ — 0436e0f

Commits on Oct 8, 2020

fix pvs check
filipecaixeta committed 14 hours ago ✕ — 3861a03

Some checks were not successful

2 failing, 5 pending, and 8 successful checks

✕	**PVS check** — Found 2 new errors, total 30 errors	Details
✕	**Style check** — Found 2 errors	Details
●	**ClickHouse build check** *Pending — 7/16 builds fini...*	Details
●	**ClickHouse special build check** *Pending — 4/5 bui...*	Details
●	**Functional stateful tests (debug)** *Pending — Started*	Details
●	**Functional stateless tests (debug)** *Pending — Star*	Details

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About



Add an SQL function formatRead | ClickHouse Pvs Check for PR #15

clickhouse-test-reports.s3.yandex.net/15497/146f973437b0a9d62c047efee1fd583f823501e8/pvs_check.html#fail1

ClickHouse Pvs Check for PR #15497

test_run.txt.out.log | PR #15497 | Commit | Help | Task (private network)

Test name	Test status	Test time, sec.
HTML report	Look at the report	
/repo_folder/src/Functions/FunctionsFormatting.h:386	FAIL	
/repo_folder/src/Functions/FunctionsFormatting.h:459	FAIL	



Add an SQL function formatRead × | ClickHouse Pvs Check for PR #15 ×

clickhouse-test-reports.s3.yandex.net/15497/146f973437b0a9d62c047efee1fd583f823501e8/pvs_check.html#fail1

ClickHouse Pvs Check for PR #15497

test_run.txt.out.log | PR #15497 | Commit | Help | Task (private network)

Test name	Test status	Test time, sec.
HTML report	Look at the report	
/repo_folder/src/Functions/FunctionsFormatting.h:386	FAIL	
/repo_folder/src/Functions/FunctionsFormatting.h:459	FAIL	



Add an SQL function formatRead | ClickHouse Pvs Check for PR #15 | PVS-Studio HTML Report

clickhouse-test-reports.s3.yandex.net/15497/146f973437b0a9d62c047efee1fd583f823501e8/pvs_studio_report/pvs-studio-h...

Group	Location	Level	Code	Message
General Analysis	XDBCDictionarySource.cpp:221	Medium	V1051	Consider checking for misprints. It's possible that the 'response' should be checked here.
General Analysis	WriteBufferFromArena.h:55	High	V1053	Calling the 'nextImpl' virtual function in the constructor may lead to unexpected result at runtime.
General Analysis	wide_integer_impl.h:35	Medium	V1061	Extending the 'std' namespace may result in undefined behavior.
General Analysis	Volnitsky.h:280	Medium	V1048	The 'chars.c1' variable was assigned the same value.
General Analysis	vdso.c:44	Medium	V707	Giving short names to global variables is considered to be bad practice. It is suggested to rename 'eh' variable.
General Analysis	UTF8Helpers.h:45	Medium	V560	A part of conditional expression is always true: first_octet < 0x80. The value range of char type: [-128, 127].
General Analysis	UTF8Helpers.h:45	Medium	V560	A part of conditional expression is always false: first_octet >= 0xF8. The value range of char type: [-128, 127].
General Analysis	Types.h:263	Medium	V1061	Extending the 'std' namespace may result in undefined behavior.
General Analysis	TwoLevelHashTable.h:169	Medium	V730	Not all members of a class are initialized inside the constructor. Consider inspecting: container, bucket, current_it.



Add an SQL function formatRead | ClickHouse Pvs Check for PR #15 | PVS-Studio HTML Report

clickhouse-test-reports.s3.yandex.net/15497/146f973437b0a9d62c047efee1fd583f823501e8/pvs_studio_report/pvs-studio-h...

General Analysis	gtest_compressionCodec.cpp:1132	Medium	V788	The variable 'prev', captured in a lambda expression, has a constant value.
General Analysis	gtest_cascade_and_memory_write_buffer.cpp:47	Medium	V522	There might be dereferencing of a potential null pointer 'wbuf_readable'.
General Analysis	GroupByFunctionKeysVisitor.h:100	Medium	V547	Expression '!keep_key' is always true.
General Analysis	greatCircleDistance.cpp:158	Medium	V614	Potentially uninitialized variable 'k_lon' used.
General Analysis	greatCircleDistance.cpp:158	Medium	V614	Potentially uninitialized variable 'k_lat' used.
General Analysis	getLeastSupertype.cpp:388	Medium	V1051	Consider checking for misprints. It's possible that the 'min_bit_width_of_integer' should be checked here.
General Analysis	FunctionsFormatting.h:459	High	V522	Dereferencing of the null pointer 'maximum_unit_const_col' might take place.
General Analysis	FunctionsFormatting.h:386	High	V547	Expression 'maximum_unit_int < 1' is always false.
General Analysis	FunctionsConversion.h:2088	High	V788	Uninitialized variable 'to_nested_type' will be used in the lambda expression, as it was captured by value.
General Analysis	FunctionsConversion.h:2088	High	V788	Uninitialized variable 'from_nested_type' will be used in the lambda expression, as it was captured by value.
General Analysis	FunctionsCoding.h:1712	Medium	V569	Transformation of constant value 128. The value range of signed char type: [-128, 127].
General Analysis	FixedHashTable.h:232	Medium	V730	Not all members of a class are initialized inside the constructor. Consider inspecting: size.



Add an SQL function formatRead | ClickHouse Pvs Check for PR #15 | PVS-Studio HTML Report

clickhouse-test-reports.s3.yandex.net/15497/146f973437b0a9d62c047efee1fd583f823501e8/pvs_studio_report/pvs-studio-h...

General Analysis	gtest_compressionCodec.cpp:1132	Medium	V788	The variable 'prev', captured in a lambda expression, has a constant value.
General Analysis	gtest_cascade_and_memory_write_buffer.cpp:47	Medium	V522	There might be dereferencing of a potential null pointer 'wbuf_readable'.
General Analysis	GroupByFunctionKeysVisitor.h:100	Medium	V547	Expression '!keep_key' is always true.
General Analysis	greatCircleDistance.cpp:158	Medium	V614	Potentially uninitialized variable 'k_lon' used.
General Analysis	greatCircleDistance.cpp:158	Medium	V614	Potentially uninitialized variable 'k_lat' used.
General Analysis	getLeastSupertype.cpp:388	Medium	V1051	Consider checking for misprints. It's possible that the 'min_bit_width_of_integer' should be checked here.
General Analysis	FunctionsFormatting.h:459	High	V522	Dereferencing of the null pointer 'maximum_unit_const_col' might take place.
General Analysis	FunctionsFormatting.h:386	High	V547	Expression 'maximum_unit_int < 1' is always false.
General Analysis	FunctionsConversion.h:2088	High	V788	Uninitialized variable 'to_nested_type' will be used in the lambda expression, as it was captured by value.
General Analysis	FunctionsConversion.h:2088	High	V788	Uninitialized variable 'from_nested_type' will be used in the lambda expression, as it was captured by value.
General Analysis	FunctionsCoding.h:1712	Medium	V569	Transformation of constant value 128. The value range of signed char type: [-128, 127].
General Analysis	FixedHashTable.h:232	Medium	V730	Not all members of a class are initialized inside the constructor. Consider inspecting: size.



Add an SQL function formatRead ✕ | ClickHouse Pvs Check for PR #15 ✕ | PVS-Studio HTML Report ✕ | FunctionsFormatting.h ✕ +

clickhouse-test-reports.s3.yandex.net/15497/146f973437b0a9d62c047efee1fd583f823501e8/pvs_studio_report/pvs-studio-h...

```
e <typename T>
ecuteType(Block & block, const ColumnNumbers & arguments, size_t result) const

ing maximum_unit = "";
(arguments.size() == 2)

 const ColumnPtr & maximum_unit_column = block.getByPosition(arguments[1]).column;
 if (const ColumnConst * maximum_unit_const_col = checkAndGetColumnConstStringOrFixedString(maximum_unit_colum
     maximum_unit = maximum_unit_const_col->getValue<String>();
 else
     throw Exception(
         "Illegal column " + maximum_unit_const_col->getName() + " of argument of function " + getName(), Erro
```

↑ V522 Dereferencing of the null pointer 'maximum_unit_const_col' might take place.

```
(const ColumnVector<T> * col_from = checkAndGetColumn<ColumnVector<T>>(block.getByPosition(arguments[0]).colum
```





HL
2020
HighLoad++

Итого

- Коммиты теперь проверяются автоматически
- Для каждой проверки можно посмотреть отдельный fullhtml-лог



Итого

Алексей Миловидов
Разработчик ClickHouse и Яндекс.Метрики

"Анализ заработал *сразу как надо*"

– Алексей Миловидов



Еще об обеспечении качества

- -Wall
- -Wextra
- -Weverything
- Сборка одновременно и gcc и clang
- Сборка с clang-tidy, проверки clang static analyzer
- Тесты с Address Sanitizer
- Memory Sanitizer
- Thread Sanitizer
- Undefined Behaviour Sanitizer
- Тестовые среды с реальными данными
- Фаззинг
- ...



Примеры найденных ошибок

Пример №1

```
if (pos + 5 <= end
  && pos[0] >= 0xC0 && pos[0] <= 0xDF
  && pos[1] >= 0x80 && pos[1] <= 0xBF
  && pos[2] >= 0x20 && pos[2] <  0x80
  && !isAlphaASCII(pos[2])
  && pos[3] >= 0xC0 && pos[0] <= 0xDF
  && pos[4] >= 0x80 && pos[4] <= 0xBF)

....
```




Пример №1

```
if (pos + 5 <= end
  && pos[0] >= 0xC0 && pos[0] <= 0xDF
  && pos[1] >= 0x80 && pos[1] <= 0xBF
  && pos[2] >= 0x20 && pos[2] <  0x80
  && !isAlphaASCII(pos[2])
  && pos[3] >= 0xC0 && pos[0] <= 0xDF
  && pos[4] >= 0x80 && pos[4] <= 0xBF)
....
```

V501 There are identical sub-expressions 'pos[0] <= 0xDF' to the left and to the right of the '&&' operator.



Пример №1

```
if (pos + 5 <= end
  && pos[0] >= 0xC0 && pos[0] <= 0xDF
  && pos[1] >= 0x80 && pos[1] <= 0xBF
  && pos[2] >= 0x20 && pos[2] <  0x80
  && !isAlphaASCII(pos[2])
  && pos[3] >= 0xC0 && pos[3] <= 0xDF
  && pos[4] >= 0x80 && pos[4] <= 0xBF)
....
```

V501 There are identical sub-expressions 'pos[0] <= 0xDF' to the left and to the right of the '&&' operator.



Пример №2

```
if (unlikely(array_size) < 0)
   throw Exception(....);
```




Пример №2

```
if (unlikely(array_size) < 0)
   throw Exception(....);
```

V562 It's odd to compare a bool type value with a value of 0: (!!(array_size)) < 0.
V547 Expression '(!!(array_size)) < 0' is always false.



Пример №2

```
if (unlikely(array_size < 0))
   throw Exception(....);
```

V562 It's odd to compare a bool type value with a value of 0: (!!(array_size)) < 0.
V547 Expression '(!!(array_size)) < 0' is always false.



Пример №3

```
const_iterator& operator ++(int) {
  const_iterator tmp = *this;
  ++*this;
  return tmp;
}
```




Пример №3

```
const_iterator& operator ++(int) {
  const_iterator tmp = *this;
  ++*this;
  return tmp;
}
```

V558 Function returns the reference to temporary local object: tmp.



Пример №3

```
const_iterator operator ++(int) {
  const_iterator tmp = *this;
  ++*this;
  return tmp;
}
```

V558 Function returns the reference to temporary local object: tmp.



Пример №4

```
if (!data_type->canBePromoted())
   throw new Exception { .... };
```




Пример №4

```
if (!data_type->canBePromoted())
  throw new Exception { .... };
```

V1022. An exception was thrown by pointer. Consider throwing it by value instead.



Пример №4

```
if (!data_type->canBePromoted())
  throw Exception { .... };
```

V1022. An exception was thrown by pointer. Consider throwing it by value instead.



Пример №5

```
if (maximum_unit_const_col)
    ....
else
    throw Exception("Illegal column "
        + maximum_unit_const_col->getName()
        + " of argument of function "
        + getName()
        , ErrorCodes::ILLEGAL_COLUMN);
```




Пример №5

```
if (maximum_unit_const_col)
  ....
else
  throw Exception("Illegal column "
    + maximum_unit_const_col->getName()
    + " of argument of function "
    + getName()
    , ErrorCodes::ILLEGAL_COLUMN);
```

V522 Dereferencing of the null pointer 'maximum_unit_const_col' might take place.



Пример №5

```
if (maximum_unit_const_col)
  ....
else
  throw Exception("Illegal column of
        argument of function "
      + getName()
      , ErrorCodes::ILLEGAL_COLUMN);
```

V522 Dereferencing of the null pointer 'maximum_unit_const_col' might take place.



Пример №5

```
if (maximum_unit_const_col)
   ....
```

V522 Dereferencing of the null pointer 'maximum_unit_const_col' might take place.



Add an SQL function formatRead ×

github.com/ClickHouse/ClickHouse/pull/15497/commits/3861a0392736b866ad77e70da39aeba8c985242b

Merged **Add an SQL function formatReadableTimeDelta to format time delta** #15497

Changes from 1 commit ▾ File filter... ▾ × Clear filters Jump to... ▾

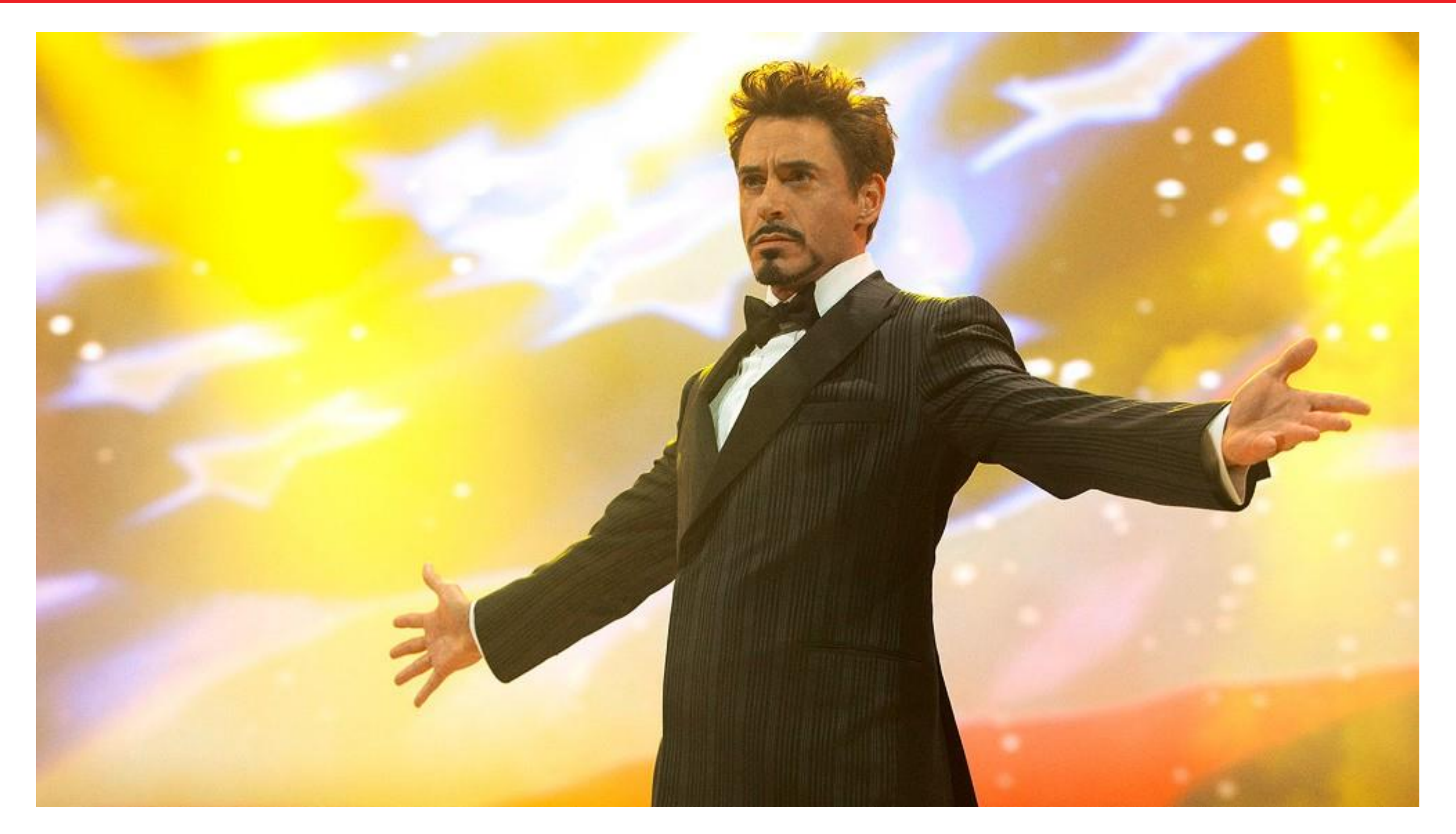
4 src/Functions/FunctionsFormatting.h

```
@@ -455,9 +455,6 @@ class FunctionFormatReadableTimeDelta : public IFunction
455             const ColumnConst * maximum_unit_const_col = checkAndGetColumnConstStringOrFixedString(maximum_unit_column.get());
456             if (maximum_unit_const_col)
457                 maximum_unit = maximum_unit_const_col->getValue<String>();
    -           else
    -               throw Exception(
    -                   "Illegal column " + maximum_unit_const_col->getName() + " of argument of function " + getName(), ErrorCodes::IL
458         }
459
460         if (const ColumnVector<T> * col_from = checkAndGetColumn<ColumnVector<T>>(block.getByPosition(arguments[0]).column.get()))
@@ -484,7 +481,6 @@ class FunctionFormatReadableTimeDelta : public IFunction
481             block.getByPosition(result).column = std::move(col_to);
482             return true;
```




Заключение

Заключение



Настроить статанализ в open-source-проекте – это не больно и не страшно!

Подарок для слушателей от PVS-Studio

Бесплатная лицензия для open-source:



www.viva64.com/pvs-free-opensource

Промокод на месяц:



www.viva64.com/pvs-download-highload


HL
++
2020
HighLoad++
